



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques i Informàtica
Universitat de Barcelona**

IMPLEMENTACIÓ D'UNA BASE DE DADES MEDIAMBIENTAL

Autora: Olga Fuentes Mañé

Director: Dr. Llorenç Servera Serapio
Realitzat a: Departament de
Matemàtiques i Informàtica
Barcelona, 31 de Gener de 2018

Resum

El projecte que es presenta a continuació es basa en la implementació d'una base de dades mediambiental.

La idea és facilitar a tothom l'accés a la informació referent a l'estat de la nostre climatologia ja que considerem que ara mateix no hi ha prou facilitat per accedir a certa informació meteorològica. Aquest projecte està enfocat a tota persona que vulgui saber com evoluciona la climatologia, però en especial a totes aquelles persones interessades en el medi ambient, les quals podran aprofitar la informació obtinguda per a les seves investigacions personals.

La base de dades estarà ubicada en una Raspberry, la qual farà la funció de servidor.

La informació a emmagatzemar a la base de dades procedeix de diferents sensors connectats cadascun d'ells a un Arduino. Aquests micro computadors envien cada tres minuts les diferents lectures dels diferents sensors a una Raspberry, un altre micro computador. La Raspberry serà l'encarregada d'emmagatzemar la informació dels diferents sensors a la corresponent taula de la base de dades. A més a més, la Raspberry tindrà una càmera connectada a ella la qual realitzarà una fotografia en el mateix moment en que les dades dels sensors són enviades.

Per a que aquesta informació sigui visible s'ha procedit a realitzar una pàgina web. En aquesta web l'usuari final pot escollir la ubicació dels sensors i també podrà escollir el tipus de sensor del qual vol veure la informació.

Per concloure s'ha comprovat que les lectures dels sensors siguin correctes. També s'ha comprovat que la informació dels diferents sensors arribi correctament a la base de dades, així com també s'ha fet un test d'usabilitat de la web per poder adaptar el seu disseny a l'usuari final.

Resumen

El proyecto que se presenta a continuación se basa en la implementación de una base de datos medioambiental.

La idea es hacer llegar a todo el mundo el estado de nuestra climatología ya que consideramos que ahora mismo no hay mucha facilidad para acceder a cierta información relacionada con la meteorología. Con lo que este proyecto va dirigido a todos, pero en especial a todas esas personas interesadas por nuestro medioambiente, las cuales luego podrán aprovechar toda la información recogida para sus propias investigaciones si así lo desean.

La base de datos estará ubicada en una Raspberry, la cual hará la función de servidor.

La información para almacenar en la base de datos procede de distintos sensores conectados cada uno de ellos a un Arduino. Estos microcomputadores envían cada tres minutos las diferentes lecturas de los distintos sensores a una Raspberry, otro microcomputador. La Raspberry será la encargada de almacenar la información de los diferentes sensores en la tabla correspondiente de la base de datos. Además, la Raspberry tendrá una cámara conectada a ella que realizará una fotografía en el mismo instante en que los datos de los sensores son enviados.

Para que esta información sea visible se ha procedido a realizar una página web. En esta web el usuario final podrá escoger la ubicación de los sensores y también podrá escoger el tipo de sensor del cual quiere obtener la información.

Para concluir se ha comprobado que las lecturas de los sensores sean correctas. También se ha comprobado que la información de los distintos sensores llegue correctamente a la base de datos, así como también se ha realizado un test de usabilidad de la web para poder adaptar el diseño de ésta el máximo posible al usuario final.

Abstract

The project presented below is based on the implementation of an environmental database.

The idea is to get the status of our climatology to the whole world since we consider that right now there is no facility to access certain information related to meteorology. With what this project is aimed at everyone, but especially to all those people interested in our environment, which will then be able to take advantage of all the information gathered for their own research if they wish.

The database will be in a Raspberry, which will have had the server role.

The information to store in the database comes from different sensors connected each one of them to an Arduino. These microcomputers send every three minutes the different readings of the sensors to a Raspberry, another microcomputer. Raspberry will be responsible for storing the information of the different sensors in the corresponding table of the database. In addition, the Raspberry will have a camera connected to it that will take a picture at the same instant that the sensor data is sent.

For this information to be visible, a web page has been made. On this website, the end user will be able to choose the location of the sensors and they can choose the type of sensor from which they want to obtain the information.

To conclude it has been verified that the readings of the sensors were correct. It has also been verified that the information of the different sensors arrives correctly to the database, as well as a web usability test that has been carried out to adapt the design of the web as much as possible for the end user.

Agraïments

Dono les gràcies a tota la meva família i amics per haver-me donat suport incondicional durant tots aquests anys.

També m'agradaria donar les gràcies als meus pares i la meva parella per haver-me aguantat en els moments més complicats.

I en especial vull dedicar aquest projecte al meu tiet Carles.

Per acabar, voldria donar les gràcies al Dr. Llorenç Servera per confiar en totes les meves propostes, i per ajudar-me i guiar-me al llarg de tot el projecte.

Sumari

Introducció	1
Estat de l'art	1
Objectius.....	3
Desenvolupament.....	4
Anàlisi de Tecnologies.....	4
Raspberry	4
Arduino	6
Raspberry PI 3 Model B	7
Frameworks	8
IDE	9
MVC	10
Comunicacions.....	11
Base de dades	13
Model relacional	15
Contingut de les taules	16
Pàgina web	20
Interacció Web.....	23
Estructura de Carpetes.....	24
Anàlisi i Cost del projecte.....	25
Anàlisi de temps	25
Cost del projecte	27
Resultats.....	29
Conclusions	32
Bibliografia	33
Webs visitades	33
Annex I -- Manual d'Usuari	35
Instal·lació bàsica.....	35
Canviar DHCP a IP Fixa.....	37
Tipus de Connexió	38
Connexió HDMI	38
Connexió VNC.....	38
Connexió SSH.....	40

Primera posada en marxa	41
Instal·lació de nous paquets.....	42
Annex II – Test Usabilitat	43
USUARI A	43
USUARI B	44
Annex III – Glossari	45

Sumari Imatges

Figura 1 – Raspberry PI 3 Model B.....	5
Figura 2 – Arduino UNO	6
Figura 3 – Esquema funcionament MVC	10
Figura 4 – Esquema de Connexions.....	11
Figura 5 – Model Relacional BDD.....	15
Figura 6 – Estructura plantilla Bootstrap	21
Figura 7 – Secció 1 de la Web.....	21
Figura 8 -- Secció 2 de la Web	22
Figura 9 – Secció 3 de la Web.....	22
Figura 10 – Dropdown	23
Figura 11 – Estructura carpetes.....	24
Figura 12 – Diagrama de Gantt	26
Figura 13 – Testing LAMP package.....	29
Figura 14 – Resultat instrucció phpinfo();	29
Figura 15 – Resultat gràfica amb Highcharts.....	30

Sumari Taules

Taula 1 – Comparativa de Raspberrys4

Taula 2 – Comparativa de Arduinos6

Taula 3 -- Comparativa entre peticions GET i POST 12

Taula 4 – Cost total del material del projecte.....27

Taula 5 – Cost total del personal del projecte.....28

Introducció

Les grans ciutats representen un pol d'atracció per a les persones que busquen noves possibilitats econòmiques i professionals, malgrat que la concentració en àrees urbanes planteja problemes de territori, sostenibilitat i recursos¹. Els episodis d'alta contaminació provocats per l'alta densitat de població i la intensa activitat econòmica, agreujats pels cada vegada més evidents efectes del canvi climàtic, són més freqüents i s'han convertit en un greu problema. Existeix la necessitat d'estudiar en profunditat aquest fenomen i analitzar les dades mediambientals per tal de trobar possibles correlacions entre elles.

La creació d'una xarxa de punts de monitorització, que aquestes dades siguin de lliure accés i no controlades per empreses o entitats, podria permetre a equips d'investigadors de qualsevol part del món poder desenvolupar models i estudiar possibles solucions fora dels interessos econòmics i buscar únicament el creixement sostenible de les grans ciutats.

Estat de l'art

Actualment podem trobar diferents pàgines webs que ens donen informació tant a temps real de l'estat de la meteorologia, com d'una previsió futura d'aquesta.

Dos exemples d'això son les webs del servei meteorològic de Catalunya [\[6\]](#) i la web de l'agència estatal de meteorologia [\[7\]](#).

En ambdues podem veure mapes a temps real de l'estat del temps. També podem descarregar-nos la informació però aquesta es descarrega en formats poc comuns i a més a més és informació ja tractada, per exemple són mitjanes anuals de precipitacions, mitjanes mensuals de temperatures, etc. Amb el que per a gent no especialitzada en la matèria es pot dir que és informació il·legible. No s'ha trobat cap però que ofereixi informació referent als diferents districtes de la ciutat de Barcelona.

Aquest projecte va sorgir amb la idea de crear una petita estació meteorològica la qual recollís diferents tipus de dades referents a la meteorologia, com la temperatura o la humitat, dades referents als nivells de radiació solar i a la qualitat de l'aire seguint un model semblant al proposat per [\[2\]](#) i que es va implementar l'any 2016 a Florència².

Tota aquesta informació, obtinguda amb els sensors esmentats, s'emmagatzema en una base de dades. Amb la informació emmagatzemada s'obtindran uns gràfics en els quals es podrà observar l'evolució de totes aquelles variables en què l'usuari estigui interessat.

¹ Mariona Tomàs Fornés, *Tendències i reptes en el govern de les gran ciutats* [\[1\]](#)

² Zaldei et al, *An integrated low-cost road traffic and air pollution monitoring platform for next citizen observatories*, *Transportation Research Procedia* 24 (2017) 531-538. [\[2\]](#)

A més a més, l'usuari tindrà la possibilitat de descarregar-se el gràfic corresponent en diferents formats. La idea és que aquestes dades siguin accessibles des d'una pàgina web la qual permeti als investigadors poder treballar amb elles i desenvolupar models³ per estudiar les possibles correlacions entre diferents variables mediambientals, especialment en episodis d'alta contaminació en ambients urbans.

Al llarg de la memòria es farà ús de paraules tècniques de les quals en podeu trobar una breu explicació al Annex III – Glossari, d'aquesta manera esperem aclarir alguns conceptes.

³ R Perez; B.M. Benito; F.J. Bonet, *ModeleR: An enviromental model repository as knowledge base for experts* [\[3\]](#)

Objectius

Els objectius proposats en aquest projecte son:

- (i) **Crear** una base de dades per a poder emmagatzemar la informació d'una petita estació meteorològica.
- (ii) **Establir** un sistema de comunicació entre els Arduino i la Raspberry.
- (iii) **Desenvolupar** una plataforma web on l'usuari pugui interactuar amb la informació meteorològica emmagatzemada a la base de dades.

La informació a emmagatzemar s'obté a partir de diferents sensors connectats cadascun d'ells a un <Arduino>.

Al llarg de tot aquest projecte tant el hardware com el software utilitzats al llarg del projecte són de tipus <Open Source>.

S'ha proposat el treballar amb plataformes hardware molt populars com són Arduino i <Raspberry>, per dos motius, són de baix preu i són fàcilment accessibles per a tothom. Per altra part es disposarà d'una ampla gamma de plaques d'expansió, anomenades <Shield>, les quals a part de facilitar-ne el seu ús ens proporcionaran noves funcionalitats.

El poder treballar amb dos hardware tant diferents com són <Arduino> i <Raspberry> és tot un repte al mateix temps que representa una important motivació. Tanmateix aquest projecte em proporciona l'opció d'aplicar molts dels coneixements adquirits al llarg de la carrera.

Desenvolupament

Anàlisi de Tecnologies

Per poder dur a terme el projecte s'ha realitzat un estudi previ dels diferents models de Raspberry que hi ha actualment al mercat per a poder fer una comparativa de les característiques de cadascun dels models i escollir el més adient.

Amb els Arduinos també s'ha realitzat un estudi dels diferents tipus existents al mercat. A més a més s'ha realitzat una comparativa de les diferents Shields d'ethernet que hi ha per a Arduinos.

Raspberry

En el cas de la Raspberry he partit de 3 punts necessaris per a realitzar el projecte. Amb el que la cerca del model de Raspberry a escollir l'he acotat bastant.

He partit de la necessitat de tenir com a mínim les següents funcionalitats a la nostra Raspberry:

- Connexió ethernet.
- Connexió per a càmera.
- Poc consum.

Tal i com s'ha comentat anteriorment he eliminat de la taula comparativa de característiques aquelles Raspberry que no tenen connexió ethernet ja que és un requisit essencial per a poder realitzar el projecte.

A continuació tenim la Taula 1 on podem veure les diferències entre les Raspberry que he escollit:

	Raspberry PI 1 Model B+	Raspberry PI 2 Model B	Raspberry PI 3 Model B
RAM	512 MB	1024 MB	1024 MB
CPU	700 MHz	900 MHz	1,2 GHz
ETHERNET	Ethernet 10/100	Ethernet 10/100	Ethernet 10/100
WIRELESS	--	--	Wifi 802.11
CAMERA	Si	Si	Si
ALIMENTACIÓ	5V 1A	5V 2A	5V 2,5A
PREU ⁴	33,50€	36,81€	37,21€

Taula 1 – Comparativa de Raspberrys

⁴ Preu obtingut del proveïdor Kubii. [\[8\]](#) el 27/12/2017

Tal i com es pot observar a la Taula 1 la gran diferència entre els models de Raspberry escollits recau en la velocitat de CPU i el tipus d'alimentació.

Per les característiques necessàries esmentades anteriorment la Raspberry escollida per a realitzar el projecte ha sigut la “Raspberry PI 3 Model B”. Podem veure a la Figura 1 la Raspberry escollida.

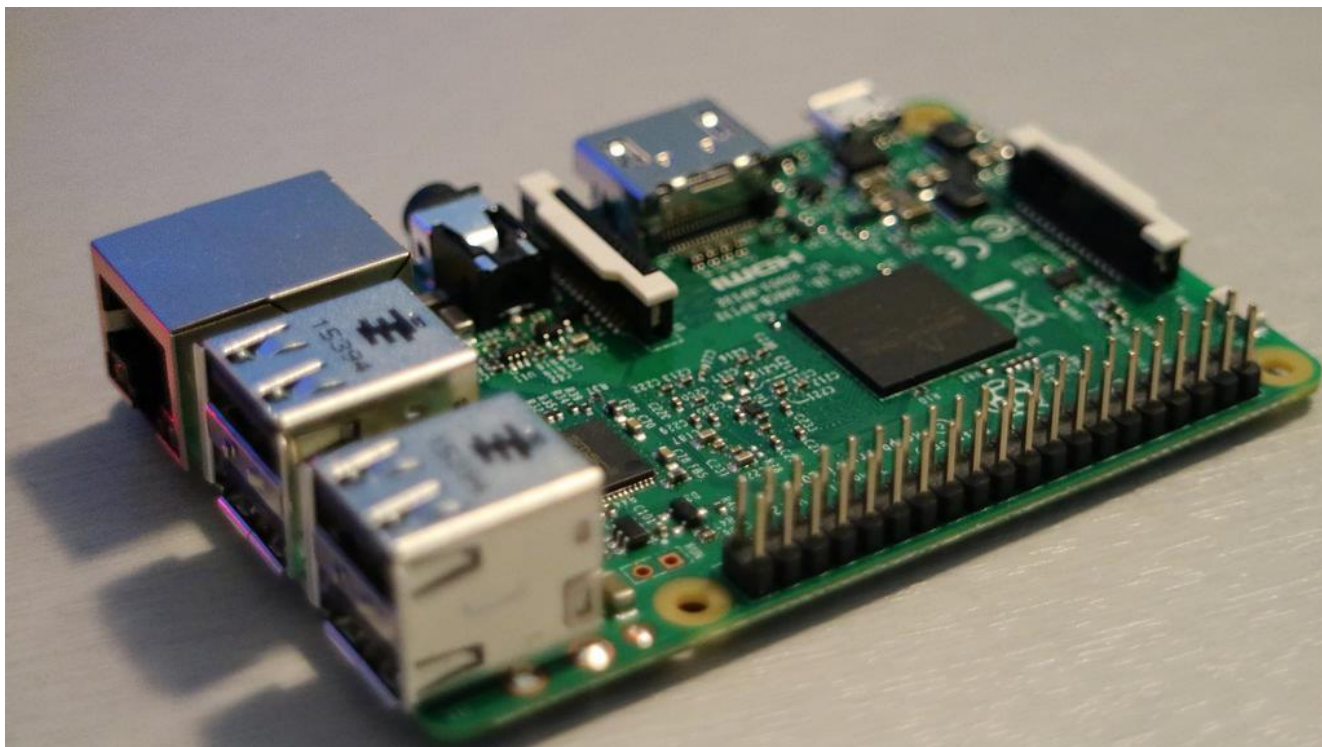


Figura 1 – Raspberry PI 3 Model B

Arduino

En el cas de les plaques Arduino podem veure la comparativa a la Taula 2.

	ARDUINO UNO	ARDUINO MEGA	GENUINO UNO	GENUINO 101	LEONARDO	ARDUINO UNO SMD
VOLTATGE	5 V	5 V	5 V	3.3 V	5 V	5 V
DIGITAL I/O PINS	14	54	14	14	20	14
ANALOG INPUT PINS	6	16	6	6	12	6
FLASH MEMORY	32 KB	256 KB	32 KB	196 KB	32 KB	32 KB
SRAM	2 KB	8 KB	2 KB	24 KB	2,5 KB	2 KB
EEPROM	1 KB	4 KB	1 KB		1 KB	1 KB
CLOCK SPEED	16 MHz	16 MHz	16 MHz	32 MHz	16 MHz	16 MHz
PREU ⁵	20 €	35 €	20 €	28,65 €	18 €	19 €

Taula 2 – Comparativa de Arduinos

A la Figura 2 podem veure l'Arduino escollit.



Figura 2 – Arduino UNO

⁵ Preu obtingut de la pàgina oficial d'Arduino [\[9\]](https://www.arduino.cc/en/Main/Store) el 27/12/2017

Raspberry PI 3 Model B

Raspberry és un microcomputador de baix cost i de petites dimensions. Aquests tipus de plaques van ser desenvolupades per a introduir la informàtica a les escoles d'una manera més amena.

Raspberry PI 3 Model B és l'última de les plaques llençades al mercat per a "Raspberry PI Foundation" amb les característiques essencials per a realitzar el projecte:

- Dimensió 13x8x4 cm
- Pes 45 gr
- Connexió ethernet
- Connexió wifi
- Connexió Bluetooth
- Port CSI per a la Càmera
- 4 ports USB
- 1 port HDMI

En el [Annex 1](#) hi podem trobar un breu manual de com realitzar una primera configuració bàsica del Sistema Operatiu a instal·lar a la Raspberry. En aquest [annex](#) també hi podem trobar un explicació dels passos a seguir per a deixar la Raspberry llesta per a poder treballar en el projecte.

Frameworks

Un Framework es pot definir com una infraestructura conceptual que ens serveix de base per al desenvolupament del nostre software.

Per definició tot framework té una arquitectura Model, Vista, Controlador. D'ara en endavant MVC.

El model gestiona les operacions lògiques.

La vista mostra el resultat a l'usuari.

El controlador gestiona el flux de dades entre el Model i la Vista. A més a més transforma les dades en funció de les necessitats de cadascun.

Al llarg del projecte s'han utilitzat diferents frameworks. Els escollits han sigut els següents:

- **LAMP**⁶
 - Linux
 - Apache
 - MySQL/MariaDB
 - PHP
- **Bootstrap** [\[10\]](#), és un framework que ens serveix per al disseny de pàgines i aplicacions web responsive. Està desenvolupat amb HTML, CSS i Javascript.
- **jQuery** [\[11\]](#), és una llibreria javascript que ens permet simplificar la manera d'interactuar amb els documents HTML.
- **Highcharts** [\[12\]](#), és una llibreria de gràfiques multiplataforma basada en SVG (Scalable Vector Graphics). Està escrita en javascript.

LAMP és el framework que hem utilitzat per a treballar des del <**Back-end**>, és a dir des de la banda del servidor.

MySQL és el gestor de base de dades que utilitzarem i serà l'encarregat d'emmagatzemar la informació provinent dels diferents sensors.

Apache és un servidor web multi plataforma de pàgines web HTTP de codi obert.

PHP és un llenguatge de programació el qual ens serveix per a poder generar pàgines web dinàmiques.

⁶ A Annex I -- Manual d'Usuari podem trobar informació de quins són els paquets necessaris a instal·lar.

Per contra Bootstrap, jQuery i HighCharts són els frameworks que treballen en el **<Front-end>**, és a dir des de la banda del client.

jQuery l'utilitzarem per a interactuar amb el **<DOM >**.

Ajax ens serveix per a realitzar comunicacions asíncrones amb el servidor. Un cop obtenim la resposta a la petició no tenim la necessitat de bloquejar ni de recarregar la pàgina.

Bootstrap ens serveix per a fer un disseny **<Responsive>** de la nostre web.

HighCharts serà l'encarregat de facilitar-nos la feina a l'hora de mostrar la informació obtinguda en els diferents sensors en forma de gràfiques.

IDE

Un IDE és un editor de text el qual té la opció de compilar i depurar el codi.

En aquest projecte he utilitzat el IDE d'editor de text SublimeText ja que és molt lleuger però a la vegada és molt potent.

MVC

A la Figura 3 podem veure un esquema general del funcionament del nostre MVC.

L'usuari final de la nostra aplicació interactua directament amb la vista.

El model serà l'encarregat de retornar al controlador el resultat de les consultes que es fan sobre la base de dades.

Un cop el controlador té aquesta informació la facilita a la vista.

Tal i com he mencionat Ajax treballa de forma asíncrona i ens evita haver de recarregar o bloquejar la pàgina en la que estigui l'usuari. Ajax també usa el MVC de forma asíncrona.

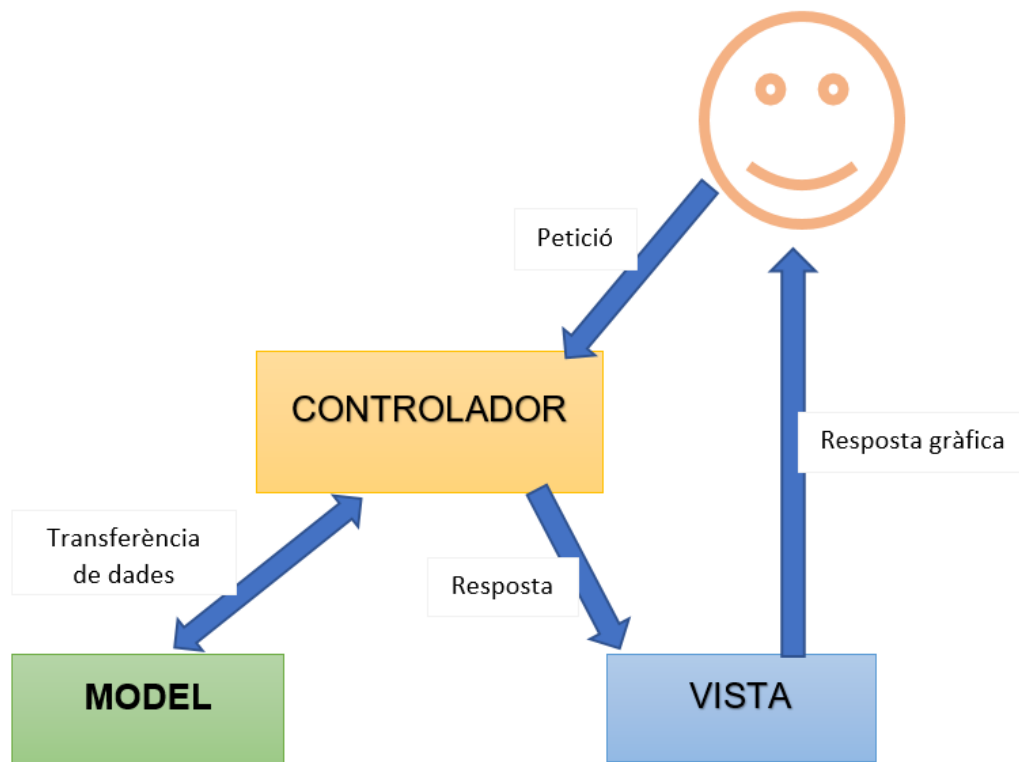


Figura 3 – Esquema funcionament MVC

Comunicacions

Una de les parts més importants del projecte és la comunicació entre els diferents Arduino i la Raspberry.

Hi ha diverses maneres de procedir.

1. Podem fer que l'Arduino estigui a l'espera i que quan rebi un senyal de la Raspberry llavors l'Arduino enviï la informació cap a la Raspberry i aquesta s'encarregarà d'emmagatzemar-ho on correspongui.
2. Podem fer que la Raspberry estigui a l'espera i que quan rebi un paquet d'un dels Arduino llavors emmagatzemarà el contingut del paquet on correspongui.

En aquest cas he optat per a la segona opció.

La Raspberry serà el nostre servidor. Aquesta estarà a l'espera a que arribi informació per a poder processar-la en cas necessari i finalment emmagatzemar-la a la taula corresponent de la base de dades.

Per altre banda tenim diferents Arduino, els quals tenen el paper de clients. Aquests enviaran cada 3 minuts la informació obtinguda per als diferents sensors a la Raspberry. Cadascun dels Arduino enviarà la informació a un arxiu en concret ubicat a la Raspberry. És dins d'aquest arxiu que la Raspberry s'encarrega d'emmagatzemar les dades a la taula corresponent de la base de dades.

Tanmateix cada 3 minuts la Raspberry realitzarà una fotografia. D'aquesta manera quan l'usuari final consulti les dades d'un tipus de sensor podrà veure addicionalment en una imatge l'estat del temps.

Per tant com acabem d'exposar l'Arduino és qui envia la informació cap al servidor.

A la Figura 4 podem veure el muntatge de connexions entre la Raspberry i els Arduino.

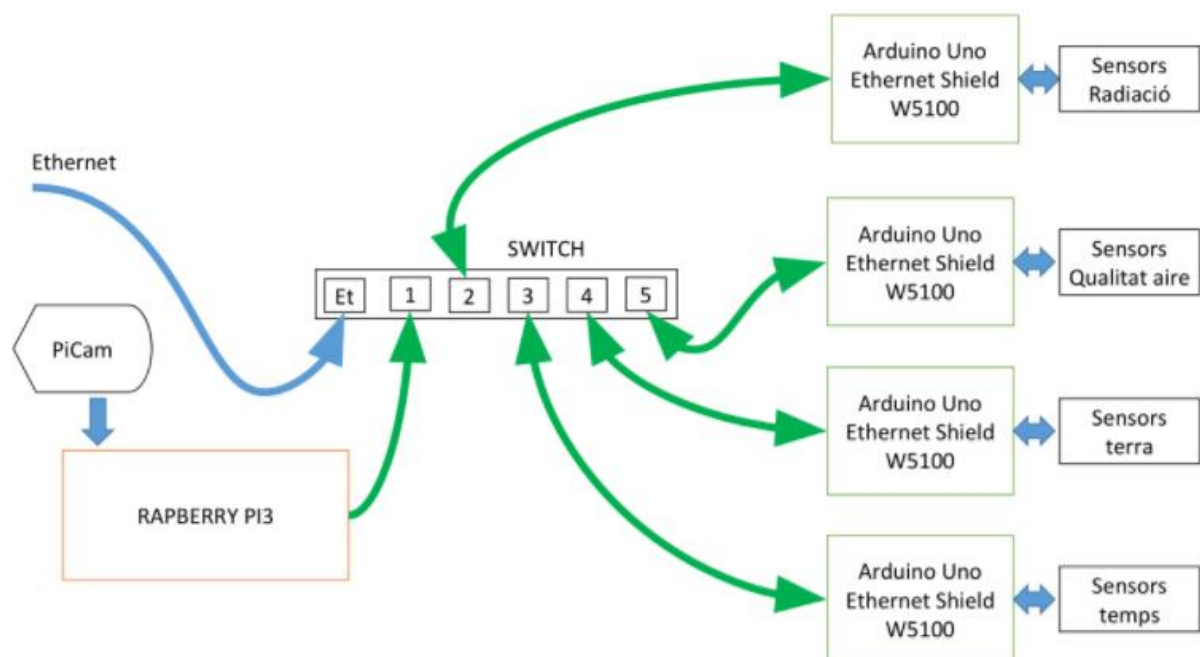


Figura 4 – Esquema de Connexions

La informació s'envia mitjançant peticions POST cap a la Raspberry.

He optat per fer la comunicació amb peticions POST ja que són més segures que les peticions GET.

A la Taula 3 podem veure una breu comparativa del comportament que tenen els dos tipus de peticions mencionades.

	GET	POST
Historial Navegador	Els paràmetres enviats es mantenen a l'historial ja que formen part de la <URL> .	Els paràmetres no es guarden a l'historial.
Visibilitat dades	La informació serà visible per a tothom ja que aquesta es mostrarà a la URL.	La informació no serà visible ja que no es mostrarà a la URL.
Restriccions de longitud	La URL té un límit d'uns 2048 caràcters, en funció del navegador.	No hi ha límit de caràcters.
Seguretat	És menys segur ja que la informació que s'envia forma part de la URL, amb el que aquesta es guardarà a l'historial del navegador i als logs del servidor.	És més segur ja que la informació que s'envia no es guarda a l'historial del navegador ni als logs del servidor.
Hacked	Més fàcil de hackejar.	Més difícil de hackejar.

Taula 3 -- Comparativa entre peticions GET i POST

Base de dades

La base de dades és el nucli d'aquest projecte.

Com ja he explicat anteriorment el sistema de gestió de base de dades relacional utilitzat és MySQL.

S'ha decidit utilitzar MySQL ja que el volum d'informació que s'espera tenir no és molt gran, per la qual cosa per al projecte en qüestió, no ens cal utilitzar altres softwares de gestió de base de dades com per exemple pot ser MongoDB⁷.

La base de dades és el lloc on s'emmagatzema tota la informació procedent dels diferents sensors i també és l'encarregada de proporcionar la informació que l'usuari demana plasman-la a la pàgina web, com ja s'ha mencionat anteriorment.

Per a realitzar l'estructura final de la base de dades ens varem plantejar una sèrie de preguntes:

- Què volem emmagatzemar?
- Tota estació meteorològica disposarà sempre dels mateixos sensors?
- Un mateix sensor pot existir en dos ubicacions diferents?
- Etc.

Un cop obtingudes les diferents respostes vam veure que en el cas que ens influeix en el projecte la nostra base de dades tindrà l'estructura que es pot observar a la Figura 5.

En tot moment al crear l'estructura de la base de dades he tingut en compte que quan es llença una aplicació a producció no es pot tornar a començar de zero, ja que un cop la base de dades tingui informació ens serà complicat refer l'estructura de la mateixa amb facilitat.⁸

A la Figura 5 podem veure que les taules dataair, dataradiation, datasoil i dataweather tenen una relació directa amb la taula sensorlocation. On cadascuna de les taules mencionades fins ara relacionades amb sensorlocation són les 4 tipologies de sensors que el client ha desitjat.

Aquesta relació ens permet tenir un control d'on estan ubicats els diferents sensors de totes les localitzacions existents.

La taula sensorlocation pot tenir de 1 fins a N sensors de tipus dataair, de 1 fins a N sensor de tipus dataradiation, de 1 fins a N sensors de tipus datasoil i de 1 fins a N sensors de tipus dataweather, sent un valor $2 \leq N$.

⁷ Sistema de base de dades NoSQL Per a més informació podeu consultar el següent enllaç [\[20\]](#).

⁸ Fragment extret del llibre '*Base de dades Cartesianes*' (Franquesa i Niubó, Bañeres, & Universitat de Barcelona, 2015, p. 337) [\[5\]](#)

També podem veure com la taula `sensorlocation` té relació amb les taules `sensortype` i `locations`.

La primera relació ens permet tenir definits tots els tipus de sensors que hi pot haver. Un `sensorType` pot estar N cops a la taula `sensorlocation`.

La segona relació ens permet tenir definides totes les ubicacions ja siguin ubicacions actives o no. Una localització pot estar representada N vegades a la taula `sensorlocation`.

Com a cas especial tenim la taula `Users` que no té cap relació directa amb cap de les taules mencionades anteriorment.

Aquesta taula `Users`, en un futur, serà l'encarregada d'emmagatzemar les credencials d'accés dels diferents usuaris que hagin d'accedir al back-end.

Model relacional

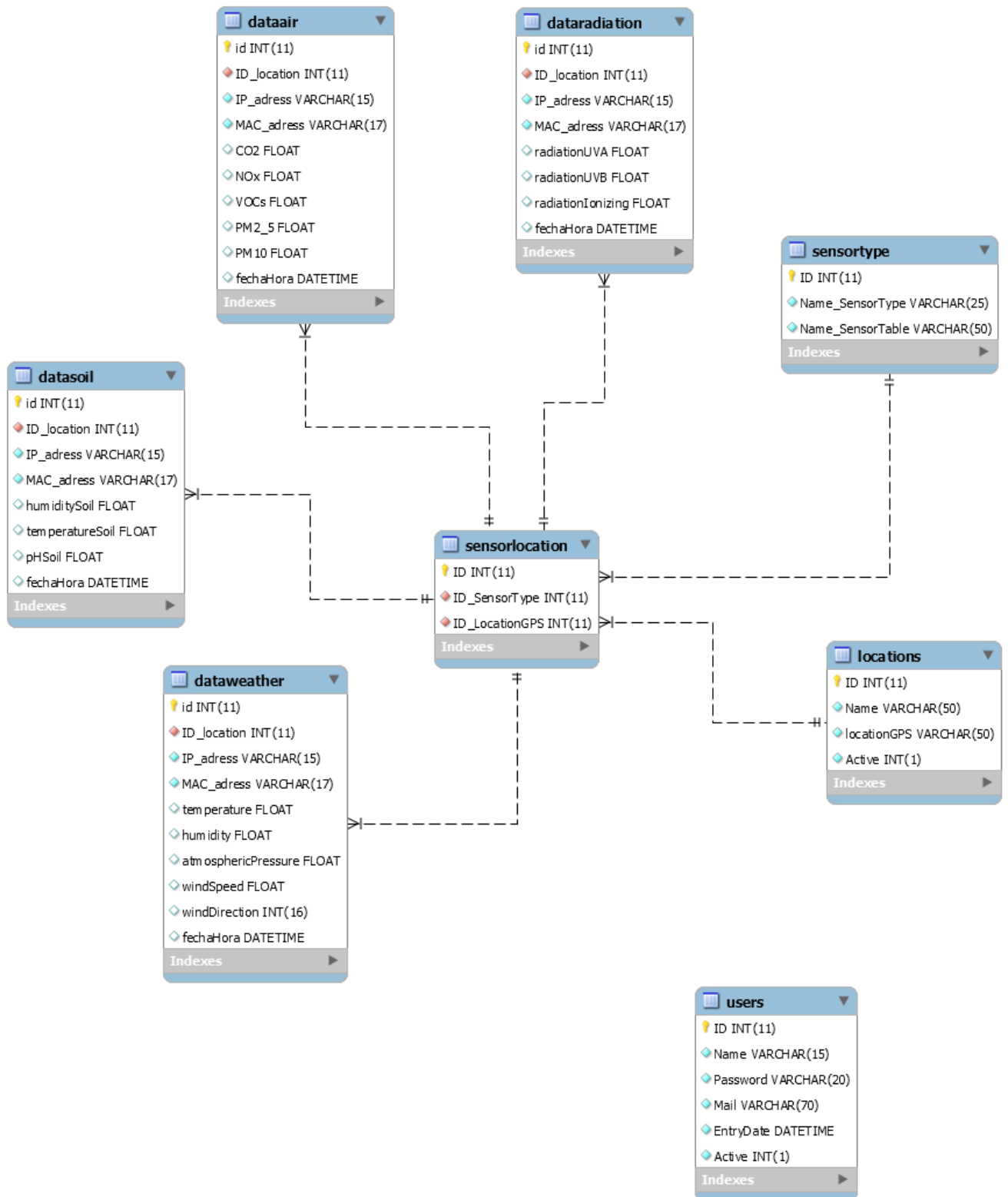


Figura 5 – Model Relacional BDD

Contingut de les taules

A continuació podem trobar una explicació de la informació que s'emmagatzema a cadascuna de les taules de la base de dades.

- **dataair** → En aquesta taula s'emmagatzemaran totes les dades dels sensors que recullin informació sobre l'estat de l'aire.
 - id: Identificador amb valor auto incremental.
 - ID_location: Aquest identificador fa referència a la localització on es troba el sensor. Aquest valor serà el del camp "id" de la taula "sensorlocation".
 - IP_adress: Adreça IP que té assignada la placa de l'Arduino d'on provenen les lectures dels sensors.
 - MAC_adress: Adreça MAC que té assignada la placa de l'Arduino d'on provenen les lectures dels sensors
 - CO2: Aquest valor representa la quantitat de CO2 que hi ha a l'aire, és a dir de la quantitat de diòxid de Carboni que es detecta.
 - NOx: Aquest valor representa la quantitat de NOx que hi ha a l'aire, és a dir de la quantitat de òxid de Nitrogen que es detecta.
 - VOCs: Aquest valor representa la quantitat de VOCs que hi ha a l'aire, és a dir la quantitat de compostos orgànics volàtils que es detecten.
 - PM2_5: Aquest valor representa la quantitat de PM2_5 que hi ha a l'aire, és a dir la quantitat de partícules en suspensió que es detecten de 2,5 micras de tamany.
 - PM10: Aquest valor representa la quantitat de PM10 que hi ha a l'aire, és a dir la quantitat de partícules en suspensió que es detecten de 10 micras de tamany.
 - fechaHora: Timestamp en el qual és emmagatzemada la informació provinents dels diferents sensors a la base de dades.

- **dataradiation** → En aquesta taula s'emmagatzemaran totes les dades dels sensors que recullin informació relacionada amb la radiació en el medi.
 - id: Identificador amb valor auto incremental.
 - ID_location: Aquest identificador fa referència a la localització on es troba el sensor. Aquest valor serà el del camp "id" de la taula "sensor_location".
 - IP_adress: Adreça IP que té assignada la placa de l'Arduino d'on provenen les lectures dels sensors.
 - MAC_adress: Adreça MAC que té assignada la placa de l'Arduino d'on provenen les lectures dels sensors
 - radiationUVA: Aquest valor representa la quantitat de radiació ultraviolada de tipus A que es detecta.
 - radiationUVB: Aquest valor representa la quantitat de radiació ultraviolada de tipus B que es detecta
 - radiationIonizing: Aquest valor representa la quantitat de radiació ionitzant que es detecta.
 - fechaHora: Timestamp en el qual és emmagatzemada la informació provinents dels diferents sensors a la base de dades.

- **datasoil** → En aquesta taula s'emmagatzemaran totes les dades dels sensors que recullin informació sobre l'estat del sòl.
 - id: Identificador amb valor auto incremental.
 - ID_location: Aquest identificador fa referència a la localització on es troba el sensor. Aquest valor serà el del camp "id" de la taula "sensor_location".
 - IP_adress: Adreça IP que té assignada la placa de l'Arduino d'on provenen les lectures dels sensors.
 - MAC_adress: Adreça MAC que té assignada la placa de l'Arduino d'on provenen les lectures dels sensors
 - humiditySoil: Aquest valor representa la humitat detectada en el sòl.
 - temperatureSoil: Aquest valor representa la temperatura del sòl.
 - pHSoil: Aquest valor representa la quantitat de pH del sòl.
 - fechaHora: Timestamp en el qual és emmagatzemada la informació provinents dels diferents sensors a la base de dades.

- **dataweather** → En aquesta taula s'emmagatzemaran totes les dades dels sensors que recullin informació sobre l'estat del temps.
 - id: Identificador amb valor auto incremental.
 - ID_location: Aquest identificador fa referència a la localització on es troba el sensor. Aquest valor serà el del camp "id" de la taula "sensor_location".
 - IP_adress: Adreça IP que té assignada la placa de l'Arduino d'on provenen les lectures dels sensors.
 - MAC_adress: Adreça MAC que té assignada la placa de l'Arduino d'on provenen les lectures dels sensors
 - temperature: Aquest valor en indica la temperatura ambient.
 - humidity: Aquest valor ens indica la humitat que hi ha a l'ambient.
 - atmosphericPressure: Aquest valor ens indica la pressió atmosfèrica existent.
 - windSpeed: Aquest valor ens indica la velocitat del vent.
 - windDirection: Aquest valor ens indica la direcció del vent.
 - fechaHora: Timestamp en el qual és emmagatzemada la informació provinents dels diferents sensors a la base de dades.

- **sensor_location** → En aquesta taula s'emmagatzemarà la relació entre un tipus de sensor i la localització de l'estació meteorològica que contindrà els diferents sensors.
 És a dir, podrem tenir un historial d'on han estat els diferents tipus de sensors.
 - id: Identificador amb valor auto incremental.
 - ID_sensorType: Amb aquest camp fem referència al tipus de sensor.
 - ID_LocationGPS: Amb aquest camp fem referència a la localització de l'estació meteorològica.

- **locations** → En aquesta taula s'emmagatzemarà una llista de totes les localitzacions on hi ha hagut sensors.
 - id: Identificador amb valor auto incremental.
 - name: Aquest valor farà referència al nom que nosaltres vulguem posar-li a la localització del GPS. Aquest nom serà el que es mostrarà posteriorment a la pàgina web.
 - locationGPS: Aquest valor farà referència a les coordenades GPS on es trobi ubicada l'estació meteorològica corresponent.
 - Active: Aquest valor ens ajudarà per a poder saber si una ubicació està activa o no. És a dir, si hi ha una estació meteorològica en funcionament o no. Si aquest camp està a false això farà que no sigui visible per a l'usuari final.

- **sensortype** → En aquesta taula s'emmagatzemarà una llista de tots els tipus de sensors definits.
 - id: Identificador amb valor auto incremental.
 - Name_sensorType: nom que tindrà el sensor.
 - Name_sensorTable: nom de la taula corresponent on el sensor emmagatzemarà la informació.

- **users** → En aquesta taula s'emmagatzemaran els usuaris que tindran accés en el futur <Back-end>.
 - id: Identificador amb valor auto incremental. Representa el numero d'usuari.
 - name: Nom de l'usuari
 - password: Contrasenya de l'usuari. Aquesta serà guardada ja parsejada amb md5.
 - email: correo de l'usuari.
 - EntryDate: data en la qual l'usuari és donat d'alta al sistema.
 - Active: Indiquem si l'usuari està actiu o no. És a dir, si pot fer login o no.

Pàgina web

En aquest punt del projecte ha sigut on he creat una pàgina web mitjançant els frameworks de Bootstrap i jQuery.

La pàgina web en qüestió és responsive. Això implica que serà visible correctament en diferents plataformes de diferents mides, com poden ser els telèfons mòbils, els ordinadors i les tauletes.

Aquesta web està basada en una de les plantilles que ofereix bootstrap gratuïtament⁹. La plantilla escollida està dividida en 3 grans blocs tal i com es pot veure a la Figura 6.

Per un costat tenim la capçalera o header, on es pot veure el logotip de la universitat, el nom de la plataforma web i les diferents seccions en les quals estarà dividit el segon bloc. El cos de la web.

Aquest segon bloc, que anomenem cos de la web o body, conté 3 seccions.

- Secció 1
 - Títol: Nosaltres
 - En aquesta secció es fa una breu descripció del projecte.
- Secció 2
 - Títol: Estacions Meteorològiques
 - En aquesta secció és on l'usuari final podrà veure en unes gràfiques el resultat de les mesures obtingudes per als diferents sensors.
- Secció 3
 - Títol: Contacte
 - En aquesta secció facilitem un correu de contacte per si l'usuari té algun dubte o suggeriment.

El tercer i últim bloc és el peu de pàgina o footer. En aquest peu de pàgina tenim la informació de la data en la que la pàgina va ser creada i informem de que la pàgina és creada amb codi lliure.

Per aconseguir un resultat final el més visualment correcte de la web he realitzat un parell de test d'usabilitat per corregir alguns possibles errors de visualització. Aquests tests els podeu trobar a Annex II – Test Usabilitat.

⁹ Podeu trobar la plantilla utilitzada en el següent enllaç [\[10\]](#)

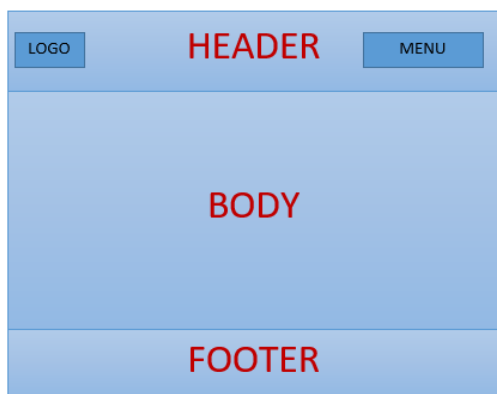
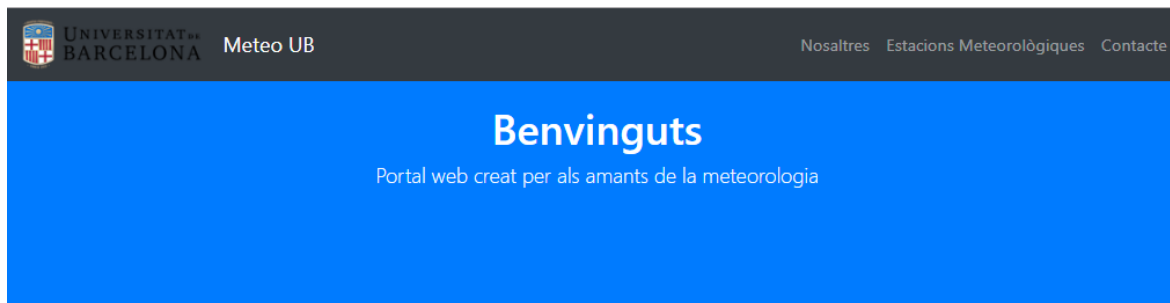


Figura 6 – Estructura plantilla Bootstrap

Podem veure a la Figura 7, a la Figura 8 i a la Figura 9 el resultat de la pàgina web. Aquesta primera Figura 7 fa referència a la Secció 1 que he esmentat prèviament.



Sobre el projecte

La principal idea d'aquest projecte és crear una base de dades mediambiental la informació de la qual estigui a l'abast de tothom.

Es vol recaptar dades mitjançant els diferents sensors que es llisten a continuació:

- Sensors de Temps
- Sensors d'Aire
- Sensors de Radiació
- Sensors de Terra

Figura 7 – Secció 1 de la Web

La Figura 8 fa referència a la Secció 2, on podem veure el resultat de consultar les dades obtingudes per al sensor de Terra. Just a sota de la gràfica amb les dades obtingudes podem veure 3 imatges. Aquestes imatges aniran variant en el temps.

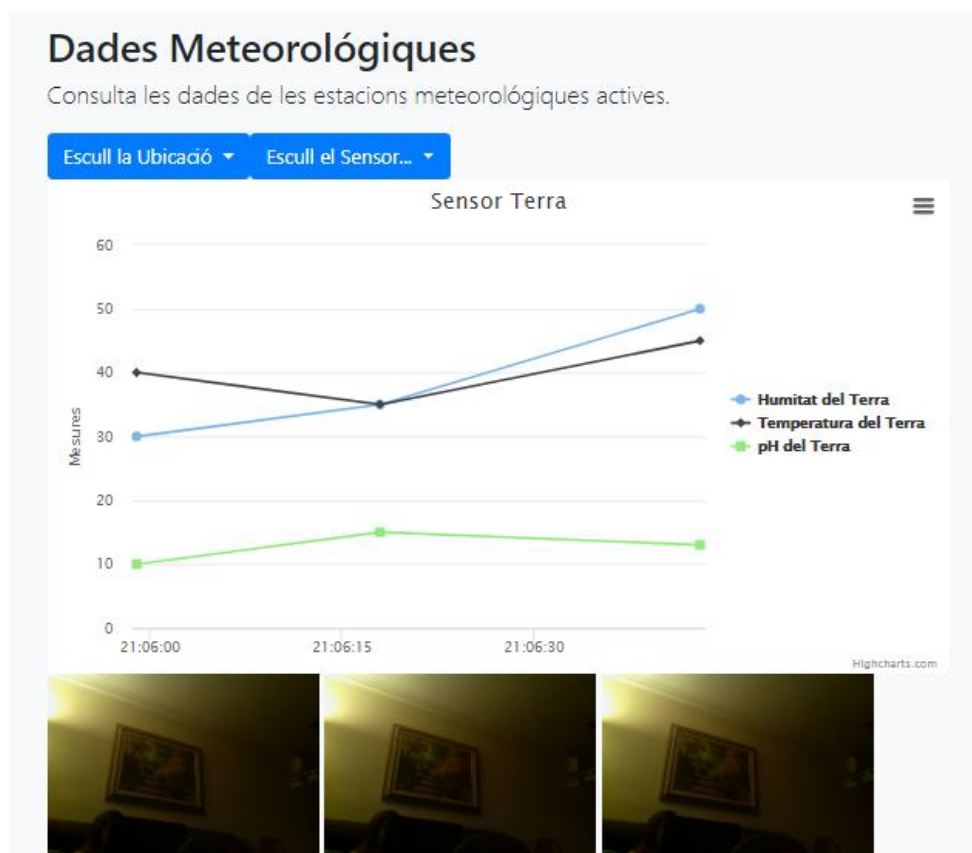


Figura 8 -- Secció 2 de la Web

En aquesta ultima Figura 9 fa referència a la Secció 3 mencionada anteriorment.

Informació de contacte

Si tens cap dubte pots contactar amb nosaltres a meteo.info@ub.edu

Figura 9 – Secció 3 de la Web

Interacció Web

Per a realitzar les consultes de les lectures dels sensors no cal que l'usuari estigui donat d'alta, de fet la pàgina web no té cap sistema de login ja que el client no ho ha trobat necessari.

L'usuari per a poder consultar les lectures desitjades haurà de fer ús dels 2 botons desplegable, anomenats dropdown.

A la Figura 10 podem veure els dos dropdowns que l'usuari haurà d'utilitzar per a escollir les lectures a obtenir. El primer fa referència a les ubicacions dels sensors i l'altre fa referència als tipus de sensors existents a la ubicació escollida.



Figura 10 – Dropdown

Cal dir que inicialment el segon botó estarà sempre deshabilitat ja no es podrà escollir un tipus de sensor sense haver escollit prèviament una ubicació.

Un cop escollides les opcions desitjades en ambdós botons es mostrarà una gràfica amb les dades corresponents. Aquesta gràfica està implementada amb la llibreria de Highcharts [\[12\]](#).

Internament tenim els següents arxius que seran els encarregats de controlar quina informació es mostra a l'usuari.

- **conectarBD.php** → En aquest arxiu tenim la configuració necessària per a realitzar la connexió amb la base de dades corresponent. Aquesta part del codi la he separat en un únic arxiu ja que quan es vulguin fer noves instal·lacions de bases meteorològiques modificant només el contingut d'aquest arxiu tota la resta funcionarà a la perfecció.
- **calculosBD.php** → En aquest arxiu tenim les diferents consultes a base de dades.
- **peticionesBD.php** → en aquest arxiu tenim les respostes possibles que retornarem a la crida per javascript feta des de l'arxiu principal.
- **index.php** → arxiu principal on tenim el codi de la pàgina web.

Estructura de Carpetes

L'estructura de carpetes de la pàgina web és la que podem veure a la Figura 11.

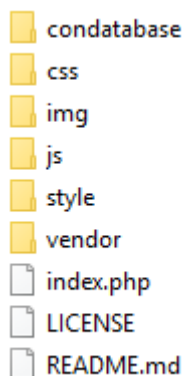


Figura 11 – Estructura carpetes

- condatabase: conté tots els arxius que accedeix a bases de dades per a realitzar les consultes necessàries.
- css: conté els arxius que donen estil a la pàgina web.
- img: conté les imatges que apareixen a la web.
- js: conté els arxius javascript que utilitzen les diferents llibreries.
- style: conté un arxiu javascript que dona format a la web.
- vendor: conté els codis dels diferents frameworks així com les dependències utilitzades per cadascun d'ells.
- LICENSE: aquest arxiu de text és informatiu. S'explica el tipus de llicència que s'usa.
- README.md: arxiu en format **<Markdown>**. En aquest arxiu hi ha una explicació del contingut del projecte. La idea és que quan hi hagi una nova funcionalitat a la web aquest canvi es vegi reflectit aquí, amb el que acabarem tenen un historial tant dels canvis importants (**<Major Update>**) com dels petits canvis per possibles errors o petites millores que sorgeixin (**<Minor Update>**).

Els usuaris finals només veuran diferències quan hi hagi una major update.

Anàlisi i Cost del projecte

Anàlisi de temps

El projecte ha tingut una durada aproximada d'uns 6 mesos, amb una dedicació diària d'unes 3-4 hores.

El temps invertit en realitzar el projecte el podem dividir en 4 grans grups.

1. Idea Inicial

En aquesta primera part del projecte hi he invertit aproximadament un terç del temps de tot el projecte.

Ens aquesta fase ha constatat en llegir tota la documentació necessària referent a les tecnologies amb les quals he treballat.

També he invertit part del temps en formació en els llenguatges de programació necessaris.

2. Disseny

En aquesta part del projecte he generat el disseny de l'estructura de la base de dades.

3. Desenvolupament

En aquesta part del projecte he generat la base de dades, la comunicació amb els Arduino i he creat el <Front-end> on mostrar els resultats obtinguts.

En aquest punt del projecte paral·lelament he anat generant la documentació per a la guia d'usuari i he anat emmagatzemant totes les pàgines visitades de les qual he obtingut alguna informació important per a dur a terme el projecte.

4. Documentació

En aquesta ultima part del projecte he fet la redacció de la memòria i he acabat la guia d'usuari per a fer la primera instal·lació i configuració de la Raspberry.

En el diagrama de Gantt que podem veure a la Figura 12 podem observar els temps associats a cadascun dels grups mencionats així com les reunions realitzades amb el tutor al llarg del projecte.

Mencionar que hi ha hagut algun bloquejant temporal però que ha afectat poc per a obtenir els resultats desitjats.

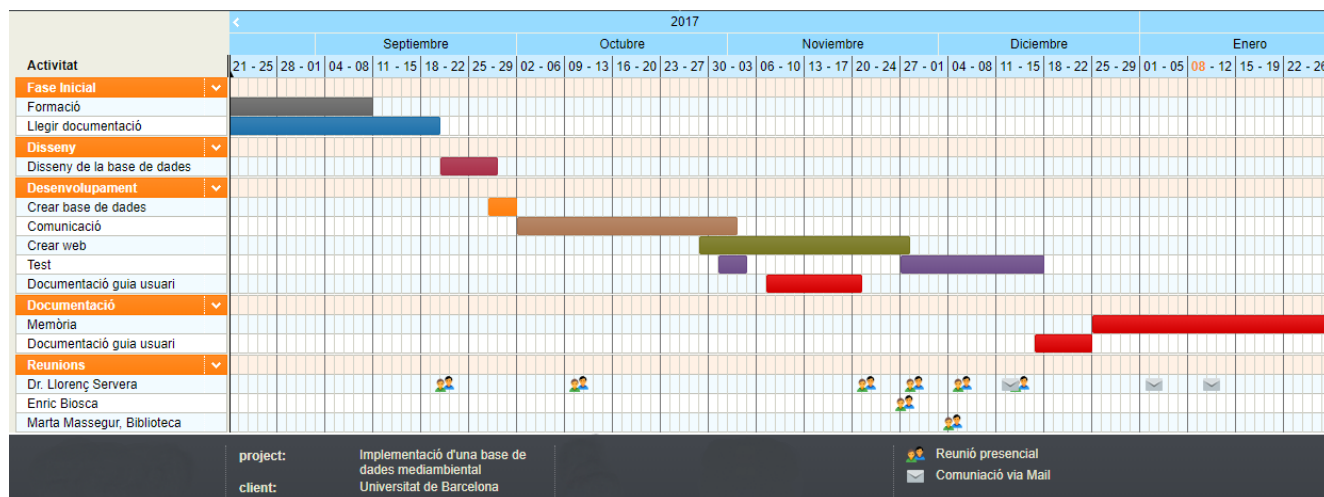


Figura 12 – Diagrama de Gantt

Cost del projecte

A la Taula 4 podem veure el cost del diferent hardware que he utilitzat per al projecte.

El preu total que es mostra en aquesta taula és el preu que costa muntar una estació meteorològica amb 1 servidor (Raspberry) amb una càmera connectada a ell, amb 4 Arduino i amb 4 tipus de sensors connectats cadascun d'ells a un dels Arduinos.

COST HARDWARE			
Concepte	Quantitat	Preu Unitat	Cost Total
Raspberry PI 3 Model B	1	37,21 €	37,21 €
Arduino UNO	4	20 €	80 €
Shield Ethernet	4	5,38 €	21,52 €
Switch TP-Link model TL-SF1005D	1	10,80 €	10,80 €
Camara	1	15 €	15€
Cables de Xarxa 1 m	5	~0,70 €	3,5 €
Cables de Xarxa de 15m	1	~0,70 €	10,5 €
Mòdul BME 680 de BOSCH	1	25 €	25 €
Mòdul CJMCU-101 OPT101	1	3 €	3 €
Mòdul UVM-30A UV Ultraviolet Ray Detection	1	6,5 €	6,5 €
Mòdul MH-Z14A	1	15 €	15 €
Mòdul MICS-6814	1	22 €	22 €
Mòdul GP2Y1014AU0F	1	4 €	4 €
Pack Radiation Sensor Board + Geiger Tube	1	143 €	143 €
TOTAL		397,03 €	

Taula 4 – Cost total del material del projecte

A la Taula 5 podem veure el cost aproximat del projecte. Hem de tenir en compte que les hores invertides en formació i documentació inicial no les hem comptades.

COST TREBALLADOR			
Concepte	Temps invertit (Hores)	Preu Hora	Cost Total
Formació i Documentació	100 hores	--	0 €
Disseny Hardware	70 hores	25 €	1750 €
Disseny Software	53 hores	25 €	1325 €
Implementació	176 hores	25 €	4400 €
Documentació	100 hores	25 €	2500€
TOTAL	499 hores		9975 €

Taula 5 – Cost total del personal del projecte

Podem veure doncs que el cost final del projecte, comptant tant el cost del hardware com el cost del personal laboral, és d'aproximadament **10.372,03 €**.

Aquest preu final no es veu afectat per cap de les llicències dels programes usats, ja que totes elles són de programari lliure.

Resultats

Un cop he anat tenint les diferents peces de l'estació meteorològica he anat fent proves per veure que tot funcionava correctament.

El primer que he tingut funcional ha sigut la Raspberry PI 3 Model B. Aquesta l'he configurat seguint els passos explicats a Annex I -- Manual d'Usuari.

Per a comprovar que tots els paquets instal·lats funcionaven bé he realitzat les següents proves.


- Per verificar que apache, mysql i php s'han instal·lat correctament he generat un arxiu index.php dins del qual he escrit el contingut que es mostra a la Figura 13.

```
<?php phpinfo();?>
```

Figura 13 – Testing LAMP package

En obrir en el navegador l'arxiu index.php ens haurà d'aparèixer una pàgina similar a la que es mostra a la Figura 14.

PHP Version 5.6.30-0+deb8u1



System	Linux raspberrypi 4.9.35-v7+ #1014 SMP Fri Jun 30 14:47:43 BST 2017 armv7l
Build Date	Apr 14 2017 15:27:54
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies

zendengine

Figura 14 – Resultat instrucció phpinfo();

Un cop he verificat que els diferents paquets instal·lats funcionen correctament he realitzat la configuració necessària i he començat a fer test a nivell de base de dades.

Per començar he comprovat que l'usuari creat per a operar amb la base de dades pot realitzar totes les operacions dels permisos assignats: INSERT, UPDATE, DELETE i SELECT.

Com inicialment no disposàvem dels sensors per enviar les dades a la Raspberry he decidit simular l'enviament de dades. Per a poder simular l'enviament de dades he creat un arxiu en Python el qual genera un valor aleatori, l'envia al corresponent arxiu de processat de dades i aquest ho envia a la taula de la base de dades que desitgem.

He creat tants arxius com sensors pot arribar a tenir l'estació meteorològica, és a dir 4 arxius. Per veure que la informació s'emmagatzema correctament he afegit una tasca programa que s'aplicarà cada 3 minuts i executarà els diferents arxius Python.

Per verificar que tot aquest procés ha funcionat he comprovat les diferents taules verificant que hi hagués contingut.

Mentre no he tingut els sensors preparats he anat comprovant el correcte funcionament de la pagina web. Aquesta internament té diferents consultes a la base de dades. Aquestes consultes retornen la informació a mostrar en els dos dropdown que podem veure a la Figura 10. En funció dels valors escollits per a l'usuari en aquests dropdown, tindrem altres consultes que ens retornaran la informació a mostrar a les gràfiques creades amb HighChart.

Aprofitant que teníem dades aleatòries a la base de dades he pogut comprovar que tot es visualitza correctament tant en dispositius mòbils, com en tauletes, com en ordinadors, tal i com es pot veure a la Figura 15.

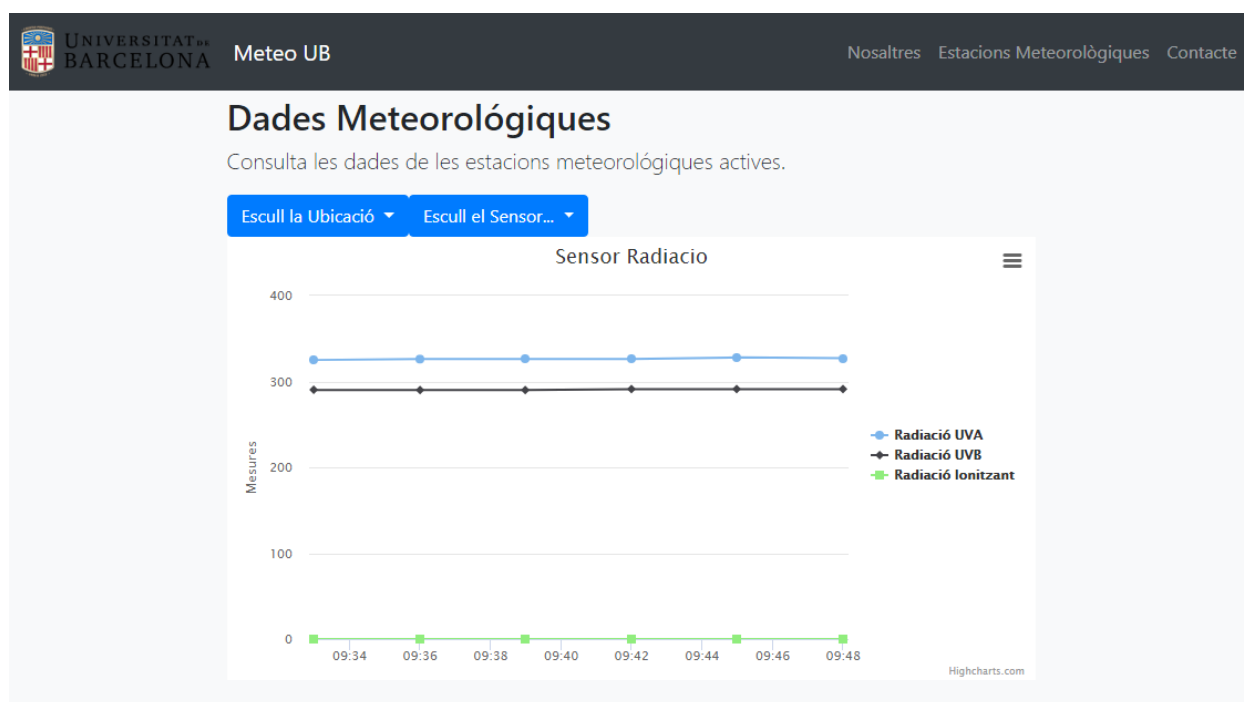


Figura 15 – Resultat gràfica amb Highcharts

Després de testejar tot el que he mencionat fins ara, i un cop he tingut els sensors preparats, he fet el muntatge que es pot veure a la Figura 4. Amb el muntatge preparat he hagut de testejar el sistema de comunicació entre la Raspberry i els Arduinos.

La prova de comunicació la he realitzat amb un Arduino, el qual tenia connectat el sensor de qualitat de l'aire que mesura les partícules en suspensió que hi ha a. Aquest Arduino un cop ha obtingut la informació del sensor ha d'enviar-la mitjançant una petició POST a l'arxiu corresponent ubicat a la Raspberry, en aquest cas a l'arxiu "getDataAir.php".

Dins d'aquest arxiu hi podem trobar el codi que recull la informació que arriba per POST i l'emmagatzema a la taula corresponent de la base de dades. En Aquest cas, l'arxiu guarda la informació a la taula "dataair".

Per comprovar que tot el procés de comunicació funciona correctament he accedit a MySQL i he consultat i verificat que la informació enviada per POST s'ha emmagatzemat a la taula de forma correcta i en el camp correcte.

A part també he accedit a la pàgina web que he creat i he comprovat que a la gràfica de dades relacionades amb la qualitat de l'aire s'hi poden veure els gràfics de la informació que acaba d'enviar el sensor a través de l'Arduino.

Quan la comunicació entre un Arduino i la Raspberry ha funcionat correctament, he procedir a reproduir els mateixos passos amb les següents proves:

- Amb diferents sensors connectats en un mateix Arduino que envien la informació obtinguda a la Raspberry.
- Amb diferents Arduinos amb varis sensors connectats a ells, que envien la informació a la Raspberry.

Conclusions

A l'inici del projecte he definit uns objectius a assolir els quals he anat explicant al llarg de la memòria.

Els objectius a assolir eren:

- (i) **Crear** una base de dades per a poder emmagatzemar la informació d'una petita estació meteorològica.
- (ii) **Establir** un sistema de comunicació entre els Arduino i la Raspberry.
- (iii) **Desenvolupar** una plataforma web on l'usuari pugui interactuar amb la informació meteorològica emmagatzemada a la base de dades.

Tal i com he explicat al llarg de la memòria per a assolir el primer dels objectius he procedit a generar la base de dades utilitzant el sistema de gestió de base de dades MySQL. Aquesta la he creat a la Raspberry que serà el nostre servidor.

La informació a emmagatzemar a la base de dades la he obtingut amb diferents sensors connectats a plaques d'Arduino.

La informació obtinguda per als sensors la he enviat mitjançant peticions POST a la Raspberry gràcies a les Shields de ethernet acoblades a les plaques d'Arduino. D'aquesta manera tenim el segon dels objectius assolit.

Per a assolir el tercer dels objectius he creat una pàgina web utilitzant el framework de Bootstrap ja que ens facilita el fet de que la web que he de creat sigui responsive.

Amb el primer dels objectius ja implementat he pogut veure, a través de la pàgina web que he creat, com la informació obtinguda per als sensors és visible en els diferents gràfics implementats amb HighCharts.

Aquest projecte té possibilitats d'ampliació.

Si s'arriba a disposar de més pressupost es poden buscar models de Shield d'ethernet més moderns o de plaques Arduino amb ethernet i wifi incorporat els quals permetin connexions HTTPS, d'aquesta manera les peticions POST amb la informació obtinguda dels sensors s'enviarà de forma encara més segura ja que podrem utilitzar certificats SSL.

Es podrien implementar noves funcionalitats, demanades per el client a últim moment, com pot ser un **<Back-end>** des d'on es puguin generar noves taules a la base de dades en calent o insertar nous valors a una taula ja existent.

Al finalitzar el projecte i després de veure totes les tasques i actuacions que comporta, i després de col·laborar amb gent d'altres sectors, puc afirmar que idealment per dur a terme un projecte d'aquestes dimensions s'ha de comptar amb un equip de gent dinàmic i especialitzat en diferents àmbits.

Bibliografia

- [1] Mariona Tomàs Fornés, Tendències i reptes en el govern de les gran ciutats.
- [2] Zaldei et al, An integrated low-cost road traffic and air pollution monitoring platform for next citizen observatories, Transportation Research Procedia 24 (2017) 531-538.
- [3] R Perez; B.M. Benito; F.J. Bonet, ModelER: An environmental model repository as knowledge base for experts. Expert System with Applications 39 (2012) 8396-8411.
- [4] Kieran Hyder, Serna Wright, Mark Kirby, Jan Brant, The role of citizen science in monitoring small-scale pollution events, Marine Pollution Bulletin 120 (2017) 51-57
- [5] Franquesa i Niubó, C., Bañeres, J., & U. D. (2015). Bases de dades cartesianes. Barcelona: Universitat de Barcelona. Publicacions i Edicions. Recollit de http://cataleg.ub.edu/record=b2137057~S1*sp

Webs visitades

- [6] <http://www.meteo.cat/prediccio/general>
- [7] <http://www.aemet.es/es/portada>
- [8] <https://www.kubii.fr/es/raspberry-pi-3-2-et-b/424-raspberry-pi-b-plus-640522710164.html?src=raspberrypi>, copyright 2016-2018
- [9] <https://store.arduino.cc> , visitada 4/09/2017
- [10] <https://startbootstrap.com/template-overviews/scrolling-nav/>, MIT License, visitada el 10/09/2017
- [11] <https://jquery.com/>, copyright 2018
- [12] <https://www.highcharts.com/> , copyright 2009-2018, Visitada 11/01/2018
- [13] <https://mobaXterm.mobatek.net/download.html>, Descàrrega del software MobaXterm, [Online], Disponible: copyright 2008-2017. Visitada 4/09/2017
- [14] <http://www.putty.org/> , visitada 4/09/2017
- [15] <https://www.w3schools.com/>, copyright 1999-2018
- [16] <https://www.python.org/>, copyright 2001-2018

- [17] <https://dev.mysql.com/doc/refman/5.7/en/datetime.html>, copyright 2018
- [18] <http://www.tightvnc.com/download.php>
- [19] <https://idescat.cat/pub/aec/236>, creative commons
- [20] <https://www.mongodb.com/what-is-mongodb>, copyright 2018
- [21] <https://bitbucket.org/product>, copyright 2018

Annex I -- Manual d'Usuari

Instal·lació bàsica

Quan tinguem ja la nostre Raspberry haurem de configurar-li un Sistema Operatiu (SO d'ara en endavant). Abans de començar amb la instal·lació del SO haurem de tenir:

- Targeta SD de mínim 8GB. (més informació¹⁰)
- 1 cable corrent
- 1 cable HDMI
- 1 ratolí amb connexió USB
- 1 cable de xarxa
- 1 teclat (opcional)

Si la targeta SD no te un SO carregat haurem de descarregar-nos-en un prèviament. En aquest enllaç¹¹ podeu trobar-hi una llista de SO per a Raspberry. En el nostre cas hem utilitzat "**Debian Jessie with Raspberry Desktop**".

Aquí deixem un enllaç¹² amb una guia de com carregar una imatge a la targeta SD.

Un cop tinguem la nostra imatge carregada correctament a la targeta SD primer procedirem a fer les connexions necessàries. Primerament haurem de connectar el ratolí, el cable HDMI el connectarem a pantalla i també connectarem un cable xarxa per a poder navegar. Quan tinguem tot aquest muntatge fet li donarem corrent.

Ara ja podem procedir a fer la instal·lació inicial del SO.

Un cop connectat a la corrent ens apareixerà una finestra similar (Figura 1) on haurem de seleccionar la opció marcada per defecte. Per qualsevol dubte deixem un enllaç¹³ a la pàgina oficial amb la guia ràpida d'instal·lació.



Figura 1 Imatge extreta de <https://www.raspberrypi.org/documentation/installation/noobs.md>

¹⁰ <https://www.raspberrypi.org/documentation/installation/sd-cards.md>

¹¹ <https://www.raspberrypi.org/downloads/>

¹² <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

¹³ <https://www.raspberrypi.org/learning/software-guide/quickstart/>

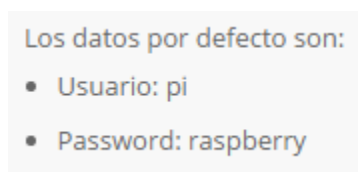
Un cop instal·lat el SO correctament ja podem procedir a configurar la Raspberry tenint en compte les necessitats del nostre projecte. Abans de desconnectar la PI haurem d'habilitar les següents opcions de connexió, totes elles des de l'apart "Configuration":

- Camera
- SSH
- VNC

A més a més si necessitem que la nostra PI tingui una adreça IP fixa podem procedir a assignar-la per tal de facilitar la feina més endavant. Per a realitzar el canvi de la IP aneu al següent apartat.

Un cop fet això ja podem reiniciar o apagar la Raspberry per a seguir amb la configuració d'aquesta.

Haurem de tenir present que l'usuari i la contrasenya que es generen per defecte al instal·lar el SO són els que es mostren a la Figura 2.

A screenshot of a text box with a light gray background. It contains the text "Los datos por defecto son:" followed by two bullet points. The first bullet point is "Usuario: pi" and the second is "Password: raspberry".

Los datos por defecto son:

- Usuario: pi
- Password: raspberry

Figura 2

Canviar DHCP a IP Fixa

En el meu cas el primer que he fet ha sigut configurar una xarxa wifi per si de cas el cable de xarxa deixés de funcionar. Per a configurar una xarxa wifi s'han de seguir els següents passos:

- Editar l'arxiu interfases. Ho farem usant la següent instrucció:

```
nano -w /etc/network/interfaces
```

En aquest arxiu el que farem és configurar la IP que desitgem per a la nostra PI. Per a poder fer la configuració necessitem saber la IP fixa que volem assignar, la màscara i la gateway. L'arxiu ha de quedar similar al que veiem a la Figura 3.

- Editar l'arxiu wpa_supplicant.conf. Ho farem usant la següent instrucció:

```
nano -w /etc/wpa_supplicant/wpa_supplicant.conf
```

En aquest arxiu el que farem és configurar la wifi on volem connectar-nos. Modificarem el camp "country" on assignarem el valor "ES" si en té un de diferent.

A continuació tal i com es veu a la Figura 4 assignarem als camps "ssid", "psk" i "key_mgmt", el nom de la nostra xarxa, la contrasenya d'accés i el tipus de connexió wifi respectivament.

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.1.175
    netmask 255.255.255.0
    gateway 192.168.1.1
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Figura 3

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=ES

network={
    ssid="          "
    psk="          "
    key_mgmt=WPA-PSK
}
```

Figura 4

Tipus de Connexió

Un cop iniciada de nou configurarem el tipus de connexió que ens vagi millor. En el meu cas he treballat directament des del terminal, però a continuació faig una breu explicació de com utilitzar les 3 connexions: SSH, VNC o HDMI.

Connexió HDMI

En aquest cas per poder treballar tan sols necessitem una pantalla amb una entrada HDMI, un cable HDMI, un teclat i un ratolí externs per connectar a la Raspberry. Això ho podem fer així ja que el SO que ens hem descarregat prèviament té una versió d'escriptori.

Connexió VNC

Si volem treballar amb un entorn gràfic però no disposem d'una pantalla o bé volem treballar a distància llavors optarem per aquesta opció.

En aquest cas haurem de descarregar-nos a l'equip des del qual volem treballar un software que permeti fer connexions remotes. En el meu cas he utilitzat TightVNC [\[18\]](#). En podeu utilitzar qualsevol altre.

Un cop tinguem aquest software descarregat i instal·lat en el nostre equip haurem de realitzar les següents instruccions des de la Raspberry.

Primer haurem d'instal·lar el següent paquet:

```
apt-get install tightvncserver
```

A continuació llençarem la següent instrucció:

```
tightvncserver
```

Aquesta instrucció ens demanarà d'introduir un password el qual necessitarem després per poder connectar-nos. En el meu cas he posat la mateixa contrasenya que té per defecte l'usuari "pi". Aquesta contrasenya s'haurà d'introduir fins a 4 cops.

Un cop fet tot això procedirem a executar el programa que hem descarregat i instal·lat en el nostre ordinador personal. Quan executem el programa ens sortirà una finestra semblant a la que veiem a la Figura 5. En aquesta finestra haurèm de posar la IP que tingui la nostra Raspberry i procedirem a fer la connexió. Si tot està bé ens apareixerà una nova finestra amb l'escriptori de la nostra Raspberry.

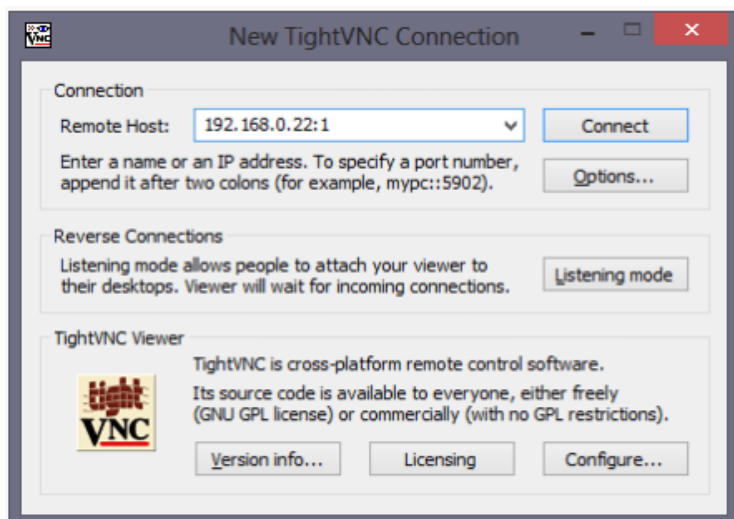


Figura 5 Finestra on passar la informació de connexió a la Raspberry

Connexió SSH

En aquest cas treballarem remotament des d'un terminal "tonto", és a dir, no tindrem cap tipus d'entorn gràfic.

Per poder connectar-nos via SSH necessitem un software com PUTTY o MobaXterm. En el meu cas he utilitzat MobaXterm ja que després d'haver provat els dos softwares considero que és més intuïtiu si mai s'ha treballat abans amb una plataforma similar.

Si no teniu el software en qüestió el podeu descarregar la versió gratuïta de MobaXterm des del següent enllaç [\[13\]](#) o si preferiu utilitzar Putty trobareu el software en aquest segon enllaç [\[14\]](#).

Un cop descarregat i instal·lat ja podem crear una nova "Putty Session".

Per a crear una nova "Putty Session" necessitem tenir la IP a la qual ens volem connectar, en aquest cas necessitem la IP que hem configurat prèviament a la nostra Raspberry, aquesta IP la haurem d'introduir el camp *Remote Host*. Pel que fa al camp del *Port* li deixarem el valor que ve per defecte.

A la Figura 6 podem veure quins camps podem omplir per a realitzar la connexió correctament.

Un cop creada la nova sessió ja podem connectar-nos remotament a la Raspberry i començar a treballar.

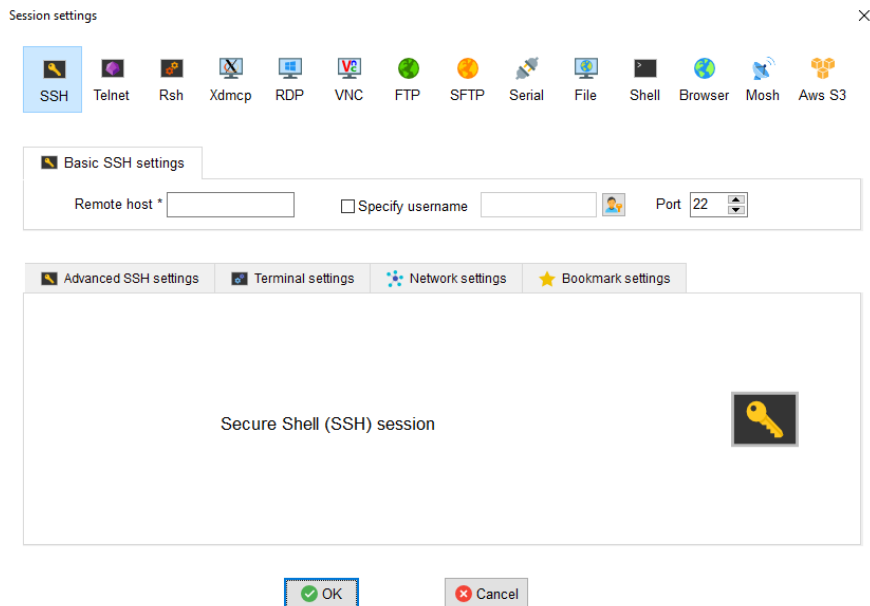


Figura 6 Finestra per crear una nova sessió remota

Primera posada en marxa

Ara que ja podem treballar des de qualsevol tipus de connexió, ja podem començar a realitzar les primeres accions bàsiques i obligatòries sobre la nostra Raspberry.

A partir d'ara donarem per suposat que estem treballant tota l'estona des d'una connexió remota realitzada des del software MobaXterm.

Un cop tinguem les sessió iniciada per a poder treballar ens haurem de registrar com a usuari root ja que és l'usuari amb tots els privilegis sobre tot el directori de carpetes. Utilitzarem la següent instrucció:

```
sudo su -
```

Primer de tot actualitzarem els repositoris dels paquets que ja venen instal·lats, amb la comanda “upgrade” i a continuació procedirem a fer un “Update” per descarregar-nos tots els paquets nous dels que disposa el repositori de la imatge que hem instal·lat.

```
apt-get -y update  
apt-get -y upgrade
```

Amb la opció “-y” forcem a que s'actualitzin tots els paquets, si no es posa la opció “-y” el sistema mostrarà la llista de canvis que es faran i tindrem l'opció d'acceptar-ho o rebutjar-ho.

Amb aquestes dues accions acabades ja podem començar a treballar.

El primer que farem és instal·lar els paquets necessaris per a poder treballar correctament. En l'apartat “Instal·lació de nous paquets” tenim la llista dels paquets a instal·lar.

El primer que instal·larem serà LAMP. Amb això ja tindrem apache, mysql i php instal·lats. Ara el que haurem de fer és configurar cadascun d'ells com nosaltres vulguem.

Referent al mysql, inicialment hem deixat les contrasenyes per defecte. Un cop hi hem pogut accedir hem creat la nostra base de dades i els usuaris necessaris per a poder-hi treballar. Cadascun d'ells amb els permisos que creguem adients. No es bona idea donar permisos totals a un usuari si aquest només ha de fer consultes sobre la base de dades.

Per evitar possibles atacs amb connexions SSH no desitjades procedirem a instal·lar el paquet del “fail2ban”. Un cop instal·lat el configurarem al nostre gust. En el nostre cas només hem modificat el temps durant el qual una adreça IP serà bloquejada en cas de detectar un accés no vàlid.

Instal·lació de nous paquets

Per a realitzar la instal·lació de qualsevol paquet utilitzarem la següent instrucció:

```
apt-get install -y "nom del paquet a instal·lar"
```

Amb l'opció "-y" el que fem es forçar la instal·lació del paquet i totes les seves dependències. Aquesta opció només la utilitzarem quan estiguem segurs de que la versió que s'instal·larà és la que ens interessa.

En aquest projecte haurem d'instal·lar com a mínim els següents paquets:

- apache2
- mysql-server-php5
- mysql
- libapache2-mod-php
- php-mcrypt
- php-mysql
- fail2ban

Annex II – Test Usabilitat

He realitzat dos test d'usabilitat a dos persones amb diferents perfils.

USUARI A

Dóna major de 45 anys.

Nivell d'estudis: Universitària.

Coneixements informàtics: Bàsics/mitjans.

Test a realitzar per la usuària.

Aquesta persona no havia tingut cap contacte anterior amb aquesta pàgina ni amb pàgines relacionades amb l'àmbit de la meteorologia.

Se li demana que primer intenti veure la gràfica del sensor de temperatura ubicat a la facultat de matemàtiques.

Una vegada obtingut el gràfic que intenti descarregar-se el gràfic en format Excel.

Un cop finalitzades les tasques anteriors se li demana que navegui per la web i que ens informi de possibles errors tipogràfics o visuals.

Resultat del test

Quan la usuària ha accedit per primer cop a la pàgina web s'ha mogut prou bé en ella.

Ha aconseguit arribar als dropdowns sense problemes i ha realitzat la consulta del sensor de temperatura ubicat a la facultat de matemàtiques sense grans problemes.

Per contra l'usuari si que s'ha encallat en el moment de descarregar-se els resultats en format Excel. Després d'un parell de minuts he hagut de donar-li una pista.

Quan ja ha realitzat els tests anteriors i ha navegat per la resta de la pàgina web, ha comentat el següent:

- Es navega bé entre apartats, és bastant intuïtiu.
- Els colors són correctes.
- Els botons on escollir la ubicació i el tipus de sensor són lletjos. Haurien de ser de color blau o negre com la resta de la pàgina.

He pres nota dels comentaris per a modificar el format del dropdown.

USUARI B

Home d'entre 25 i 30 anys.

Nivell d'estudis: Grau superior.

Coneixements informàtics: Mitjans.

Test a realitzar per l'usuari.

Aquesta persona no havia tingut cap contacte anterior amb aquesta pàgina, però si que ha visitat en algun altre moment pàgines relacionades amb l'àmbit de la meteorologia.

Se li demana que primer intenti veure la gràfica del sensor de radiació ubicat a la facultat de matemàtiques.

Una vegada obtingut el gràfic que intenti descarregar-se el gràfic en format PDF.

Un cop finalitzades les tasques anteriors se li demana que navegui per la web i que ens informi de possibles errors tipogràfics o visuals.

Resultat del test

Quan l'usuari ha accedit per primer cop a la plana web ha fet el comentari de que era una pàgina molt senzilla.

S'hi ha mogut per ella sense problemes, ha arribat a l'apartat dels dropdowns sense cap tipus de problema. Ha realitzat la consulta del sensor de radiació ubicat a la facultat de matemàtiques sense cap inconvenient. En aquest cas l'usuari s'ha pogut descarregar el resultat de la consulta en format PDF sense cap problema.

Després de deixar-lo navegar per la resta de la pàgina web l'usuari ens ha fet els següents comentaris:

- És una pàgina molt fàcil de utilitzar i de navegar.
- Els gràfics estan bé ja que es veuen amb diferents colors.
- Els botons de selecció són molt bàsics.

He pres nota per a modificar el format del dropdown.

Annex III – Glossari

En aquest annex es farà un breu explicació d'algunes paraules tècniques per a facilitar la lectura del document en qüestió.

- **Arduino:** Placa de dimensions reduïdes que es comporta com un ordinador amb un consum energètic molt inferior, però té menys prestacions que un ordinador convencional.
- **Back-end:** Pàgina web només visible per als desenvolupadors i/o administradors d'aquesta. És on es podrà fer canvis a nivell intern.
- **DOM (Document Object Model):** API on es defineix l'estructura lògica, el mode en el que s'accedeix i es manipula un element en un document HTML, XML o XHTML.
- **Front-end:** Pàgina web visible per a tothom.
- **Major Update:** Gran canvi a nivell de programació que normalment comporta canvis estructurals o de funcionament en l'aplicació. L'usuari final sí que s'adona d'aquests canvis.
- **Markdown:** Estàndard web per a afegir format als texts.
- **Minor Update:** Petits canvis a nivell de programació que no alteren l'estructura de l'aplicació ni el seu funcionament. L'usuari final normalment no se n'adona d'aquests canvis.
- **Open Source:** Dit del programari informàtic lliure de llicències de pagament.
- **Raspberry:** Placa de dimensions reduïdes que es comporta com un ordinador amb un consum energètic molt inferior.
- **Responsive:** Dit de les webs o aplicacions que tenen un disseny el qual s'adapta a diferents dispositius.
- **Shield:** Plaques d'expansió per a ampliar les prestacions de les plaques Arduino i/o Raspberry.
- **URL(Uniform Resource Locator):** Adreça específica que s'assigna a una pàgina web.